Network Embedding

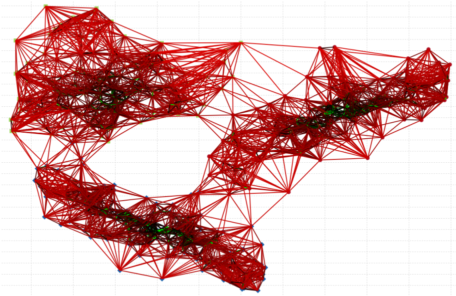
Social and Technological Networks

Rik Sarkar

University of Edinburgh, 2019.

Network Embedding

- Definition
  - Assignment of a coordinate to each node
    - f(v) gives the coordinates of node v
  - In d dimensional space
  - Usually requires unique coordinate for each vertex
- Remember: Intrinsic and extrinsic metrics
  - Intrinsic metrics: distances that can be measured purely by walking along network edges. e.g. shortest path distance
  - Exterinsic: distances between vertices in the ambient space i.e. the d-dimensional Euclidean space



Network embedding

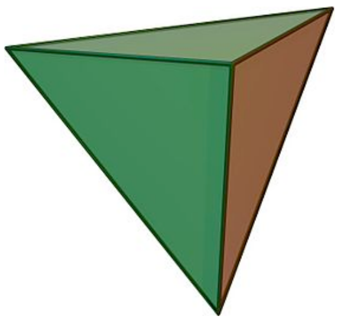
- Usually we are interested in distances between nodes (discrete)
- In some cases, points on the edges themselves may be relevant (continuous)
  - E.g. road networks

Example: suppose we want to preserve shortest path distances

- Can we embed:
  - An edge in a chain
  - A triangle in a line
  - A triangle in a 2d plane
  - A square in a 2d plane
  - A cycle in a 2d plane

Dimension Examples:

- Embedding cliques
- 1d clique: edge
- 2d clique: triangle
- 3d clique: tetrahedreon



- “simplices” (cliques) are the minimal elements of various dimensions

Tree examples:

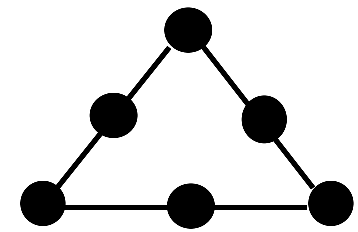
- Let’s take binary trees
- Can we embed them isometrically?
  - (while preserving all distances)

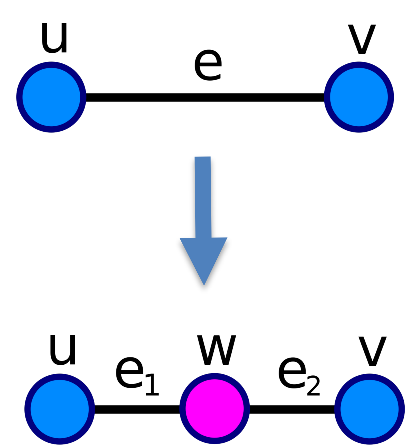
Challenges:

- Sources of problem: mismatch between intrinsic and extrinsic metrics
  - Cycles
  - Rapid branching and growth
  - High dimensions

Challenges

- Dimension of a graph is hard to characterize
- A triangle may not have 3-cliques
- Definition:
  - Subdivision: Slit an edge into two
  - Homeomorphism: Two graphs are homeomorphic if there is a way to subdivide one to get another





Challenges

- Summary: Embedding is hard
  - In general, the metric of the graph may not match with any Euclidean metric of fixed dimension. E.g. cycles, spheres, trees..
  - The right dimension d of the ambient space may be hard to decide

Theoretical results

- Smooth (See the Nash Embedding Theorem)
  - Certain classes (e.g. Riemannian manifolds of d dimension) have nice (isometric or nearly isometric) embeddings in Euclidean spaces of O(poly(d)) dimensions

- (this is a math topic. So we are stating this only vaguely. Ignore for exams.)

Distortion

- In reality, most embeddings are not perfect – they *distort* the distances
- Some distances contract, some expand
- For a metric space X with intrinsic distance d, and distance d' in the ambient (embedding space)
- Contraction: $\max_{x,y \in X} \frac{d(x,y)}{d'(f(x),f(y))}$
- Expansion: $\max_{x,y \in X} \frac{d'(f(x),f(y))}{d(x,y)}$
- Distortion = Contraction * Expansion

Distortion

- Distortion = 1 means isometric
- Nice property: Uniform scaling gives distortion = 1
  - Verify

Johnson Lindenstrauss Lemma

- A set X which is n points in k-dim Euclidean space has a an embedding in
  - Euclidean space of dim $O((\log n)/\varepsilon)$
  - with distortion at most $(1+\varepsilon)$.

- Algorithm:
  - Take $O((\log n)/\varepsilon)$ random unit vectors in R^k
  - Project (take dot product) of points of X on these vectors
  - Now we have $O((\log n)/\varepsilon)$ dim representation of X
  - Has small distortion

- This is the basis of a lot of modern data science algorithms, including compressed sensing

Random walk based node embedding

- From each node u make many random walks of length w
- Count how many times every other node occurs in these random walks N(u) (call them neighbors)
  - Estimate the probability of each nearby node occurring in these walks.
- Find embedding z, which maximizes:

$$\max_{z} \sum_{u} \log P(N(u)|z_u)$$

Given node u, predict its neighbor probabilities

Turn into a loss minimization

$$\min \mathcal{L} = \sum_{u \in V} \sum_{v \in N(u)} - \log P(v|z_u)$$

- Evaluate P as

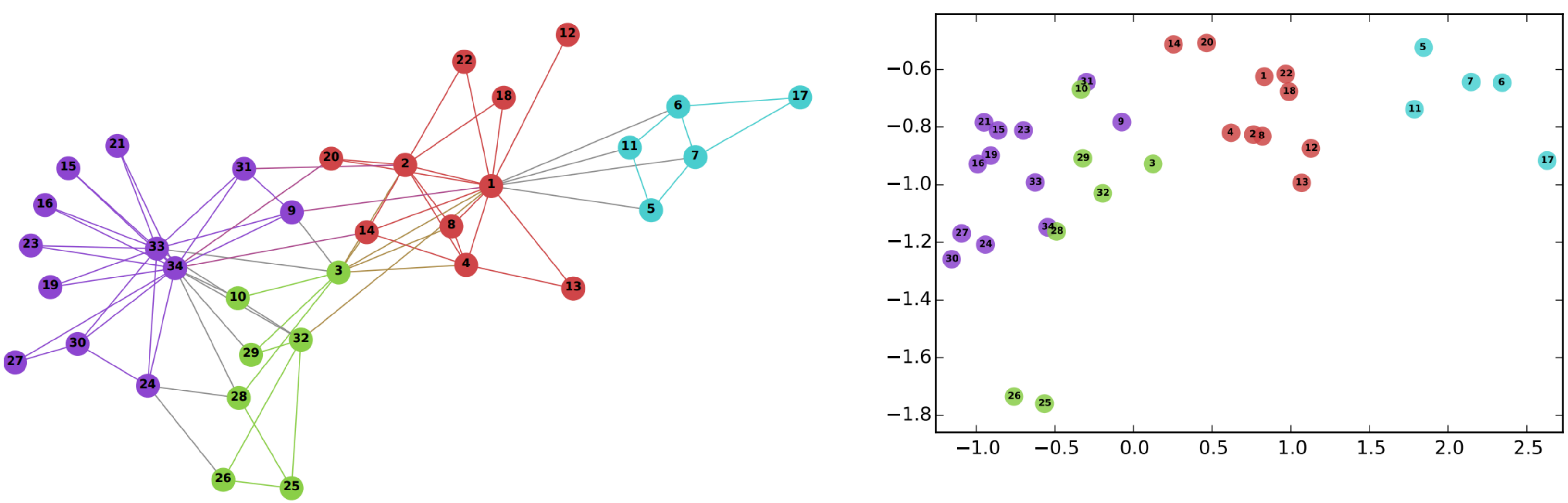
$$P(v|z_u) = \frac{\exp(z_u^T z_v)}{\sum_{n \in V} \exp(z_u^T z_n)}$$

  – Called the softmax function

Stochastic gradient descent

- The loss minimization can be done as SGD
- Take vertices in random order
  - For each z_u, take the gradient – the direction to move u to decrease loss
  - Move u slightly in the direction
- Repeat with a different random order
- Until convergence

- SGD is a standard stats technique. We will omit the details



(a) Input: Karate Graph

(b) Output: Representation

Practical considerations

- Expensive due to the $z_u^T z_n$ term that requires comparison with all vertices
- Can be approximated at a reduced cost by suitable sampling.
- SGD can be used to instead train a neural net that suggests coordinates
  - Less storage than storing all coordinates, but also less accurate
- Paper: Deepwalk. Perozi et al.
- Other variants:
  - Different ways of conducting the random walk

Applications of embedding

- Also called “representations”
- Representation learning is an important area
- Representing nodes in a Euclidean space lets us easily apply standard machine learning techniques
  - Most techniques rely on R^d Space and dot products
- Classification, clustering etc can now be performed on networks

Embedding of attributed social networks

- Suppose each node has a attributes (e.g. hobbies, interests etc)
- The ideal embedding should:
  - Represent similarity/dissimilarity of attributes
  - Represent similarity/dissimilarity of network position
- In theory, these can be opposing objective
- In practice, homophily means these are correlated

Attributed network embedding

- Minimize loss that incorporates probabilities of right neighbors as well as similar attributes

Embedding whole graphs

- Suppose there is a database of molecules
  - Each node has attributes
- We want to represent each as a points in R^d
  - Such that similar molecules are close
- Method 1:
  - Embed each as graph, then take the mean

- Method 2:
  - In each graph, perform random walks of length w starting at random points
  - Collect neighborhood sequence at each graph
  - Perform embedding so that attribute sequences seen in random walks are close

- Some authors like to distinguish as node embedding vs graph embedding

Why random walks

Why random walks

- Saves computation: no need to consider all pairs
- Known to capture relevant properties of networks like community structure
  - Highly connected nodes are likely to be close in random walks
  - Representative of diffusion processes
- First methods were inspired by NLP methods of sequences in text – random walk gives natural sequences

Embedding networks into other spaces

- Embedding into hyperbolic spaces is a popular research area these days
- Other significant papers on embedding into trees, distributions over trees etc
- Embedding can be used to compare networks
- E.g. for A and B
  - If good embeddings A -> B *and* B -> A exist, then A and B are probably similar.